

VŠB – Technická univerzita Ostrava  
Fakulta elektrotechniky a informatiky  
Katedra informatiky

# **Absolvování individuální odborné praxe**

## **Individual Professional Practice in the Company**

## Zadání bakalářské práce

Student: **Brenda Lesniczáková**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 2612R025 Informatika a výpočetní technika

Téma: Absolvování individuální odborné praxe  
Individual Professional Practice in the Company

Jazyk vypracování: čeština

### Zásady pro vypracování:

1. Student vykoná individuální praxi ve firmě: ATLAS consulting spol. s r.o.
2. Struktura závěrečné zprávy:
  - a) Popis odborného zaměření firmy, u které student vykonal odbornou praxi a popis pracovního zařazení studenta.
  - b) Seznam úkolů zadaných studentovi v průběhu odborné praxe s vyjádřením jejich časové náročnosti.
  - c) Zvolený postup řešení zadaných úkolů.
  - d) Teoretické a praktické znalosti a dovednosti získané v průběhu studia uplatněné studentem v průběhu odborné praxe.
  - e) Znalosti či dovednosti scházející studentovi v průběhu odborné praxe.
  - f) Dosažené výsledky v průběhu odborné praxe a její celkové zhodnocení.

### Seznam doporučené odborné literatury:

Podle pokynů konzultanta, který vede odbornou praxi studenta.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **RNDr. Eliška Ochodková, Ph.D.**

Konzultant bakalářské práce: Mgr. Tomáš Řehák

Datum zadání: 01.09.2018

Datum odevzdání: 30.04.2019


  
doc. Ing. Jan Platoš, Ph.D.  
vedoucí katedry



  
prof. Ing. Pavel Brandštetter, CSc.  
děkan fakulty

Prohlašuji, že jsem tuto bakalářskou práci vypracovala samostatně. Uvedla jsem všechny literární prameny a publikace, ze kterých jsem čerpala.

V Ostravě 25. dubna 2019

  
.....

Souhlasím se zveřejněním této bakalářské práce dle požadavků čl. 26, odst. 9 Studijního a zkušebního řádu pro studium v bakalářských programech VŠB-TU Ostrava.

V Ostravě 19. dubna 2019

**ATLAS consulting spol. s r.o.** (14)  
IČO 292113  
702 00 Ostrava, Moravská Ostrava  
IČO 48578706, DIČ CZ48578706

Na tomto místě bych ráda poděkovala všem kolegům z Atlas consulting, se kterými jsem měla možnost pracovat, za pomoc při práci a především Tomáši Řehákovi, za umožnění absolvování odborné praxe ve skvělém kolektivu.

## **Abstrakt**

Tato bakalářská práce je písemnou zprávou o průběhu absolvování individuální odborné praxe ve firmě Atlas consulting, spol. s r. o. V této zprávě se zaměřuji především na zpracování dat uživatelských profilů ze starších verzí informačního systému vyvíjeného společností Atlas consulting. V první kapitole popisuji odborné zaměření firmy a mé pracovní zařazení. Dále jsou popsány mně svěřené úkoly a jejich řešení. V závěru je zhodnocení dosažených výsledků mé práce.

**Klíčová slova:** Analýza dat, Python, SQL

## **Abstract**

This bachelor is report on the completion of individual professional experience, which I have done in Atlas consulting, spol. s r. o. company. In the thesis I focus exactly on processing of user profiles from older version of information system developed of Atlas consulting. In the first chapter I describe the professional orientation of the company and my work position. Further, there are described my tasks and resolutions. Finally I evaluate achieved results of my work.

**Key Words:** Data analysis, Python, SQL

# Obsah

Seznam použitých zkratk a symbolů	8
Seznam obrázků	9
Seznam výpisů zdrojového kódu	10
1 Úvod	11
2 Popis odborného zaměření firmy, u které student vykonal odbornou praxi a popis pracovního zařazení studenta	12
3 Úkoly zadané v průběhu odborné praxe	13
3.1 IQLink-reporty . . . . .	13
3.2 Zpracování uživatelských profilů . . . . .	13
4 Řešení zadaných úkolů	14
4.1 IQLink-reporty . . . . .	14
4.2 Zpracování uživatelských profilů z online verze IS . . . . .	19
4.3 Zpracování uživatelských profilů z desktop verze IS . . . . .	21
5 Závěr	23
5.1 Teoretické a praktické znalosti získané během studia uplatněné v průběhu odborné praxe . . . . .	23
5.2 Znalosti či dovednosti scházející v průběhu odborné praxe . . . . .	23
5.3 Dosažené výsledky v průběhu odborné praxe a její celkové zhodnocení . . . . .	23
Literatura	24

## Seznam použitých zkratk a symbolů

API	– Application Programming Interface
CSV	– Comma-Separated Values
ID	– IDentification
IS	– Informační Systém
JSON	– JavaScript Object Notation
REST	– Representational State Transfer
SQL	– Structured Query Language
URL	– Uniform Resource Locator



## Seznam obrázků

1	Výstup reportu . . . . .	16
2	Blok dat připravený k importu . . . . .	20
3	Firebird . . . . .	21
4	Soubor UPB se zálohou . . . . .	21
5	Záznam v Jupyteru . . . . .	21

## Seznam výpisů zdrojového kódu

1	Ukázka části skriptu pro IQLink-reports . . . . .	15
2	Ukázka zápisu uživatelských dat do souborů . . . . .	18

# 1 Úvod

Jedním z nejdůležitějších částí výuky během studia je praxe. Během ní může student uplatnit teoretické znalosti a dovednosti nabyté během studia. Rovněž má možnost čelit reálným problémům, a tím se blíže seznámit se skutečnými postupy používanými v praxi. Proto jsem se pro svou bakalářskou práci rozhodla využít možnosti absolvování individuální odborné praxe ve vybrané firmě.

K vykonání odborné praxe jsem si zvolila firmu Atlas consulting, spol. s r. o. se sídlem v Ostravě. Náplň mé práce souvisela především se zpracováním uživatelských dat, programováním v IPythonu pomocí interaktivního programu Jupyter a propojení skriptů s interními databázemi.

V této bakalářské práci nejprve představím firmu, ve které jsem odbornou praxi vykonávala. V další kapitole uvedu a stručně popíšu úkoly, které mi byly zadány. U jednotlivých úkolů se pokusím nastínit, jak jsem je řešila, jaké technologie jsem při jejich řešení používala, a s jakými problémy jsem se potýkala. V závěrečné části uvedu, které znalosti získané během studia jsem během praxe uplatnila a které mi naopak scházely.

## 2 Popis odborného zaměření firmy, u které student vykonal odbornou praxi a popis pracovního zařazení studenta

Pro vykonání odborné praxe jsem si vybrala společnost Atlas consulting spol. s r. o., jelikož mne zaujala svým odborným zaměřením i nabízenou pozicí pro vykonání praxe. Firma Atlas consulting člen skupiny ATLAS GROUP<sup>1</sup> je českou softwarovou společností se sídlem v Ostravě. Již od roku 1992 se věnuje vývoji právních a manažerských informačních systémů a aplikací.

Nejvýznamnějším produktem je bezesporu právní informační systém CODEXIS®, který je nejrozšířenějším a nejkomplexnějším právním informačním systémem v České republice. Je to komplexně řešený software navržený pro efektivní práci s informacemi ze všech oblastí práva. Je určen pro právníky a zástupce právnických profesí, finanční manažery, ekonomy, účetní, úředníky, manažery a specialisty, kteří k výkonu své práce potřebují snadný přístup k právním předpisům, zákonům a informacím k určitému tématu (např. personalistika, účetnictví, GDPR a další).

Jako jedna z mála společností nenabízela studentům jen čistě programátorské pozice, ale také pozici zaměřenou na analýzu a zpracování dat, oč jsem měla veliký zájem. Na pozici Intern data analyst jsem měla možnost spolupracovat nejen s vývojářským týmem, který nyní vyvíjí nejnovější verzi systému CODEXIS, ale i s týmy, které před lety vyvinuly první desktopovou verzi a pozdější online verzi.

---

<sup>1</sup><https://www.atlasgroup.cz/>

### 3 Úkoly zadané v průběhu odborné praxe

Během vykonávání své odborné praxe jsem měla možnost řešit několik zajímavých úkolů a získat mnoho užitečných zkušeností.

#### 3.1 IQLink-reporty

Mým prvním úkolem bylo vytvoření skriptů, které zpracovávají data o aktivitách uživatelů v IS CODEXIS. Dosavadní skripty, které využívali interní analytici, byly psané ve skriptovacím jazyce Groovy a ne všechny byly plně funkční. Skriptů je celkem 25 a výstupem každého z nich je report o některé z činností za zadané období. Například lze pomocí nich zjistit, kolik uživatelů používá desktop verzi softwaru a kolik používá online verzi, kolik uživatelů se přihlásilo, nebo kolik dokumentů bylo otevřeno, a které jsou otevírané častěji. Tyto informace přispívají ke zkvalitnění vyvíjených produktů a lepší péči o uživatele. Veškeré tyto informace se průběžně ukládají pomocí aplikace IQLink-reports ve formě JSON objektů a lze k nim přistupovat pomocí URL adresy, jejíž součástí je interval - období, jež chceme zobrazit.

Poté vznikl požadavek na vytvoření skriptu, který vytvoří report ve formě JSON objektu pro každého jednotlivého uživatele.

#### 3.2 Zpracování uživatelských profilů

V současné době se oddělení vývoje zaměřuje na vývoj a nasazení nové verze právního informačního systému CODEXIS. Ten zanedlouho úplně nahradí dosavadní online verzi a poté také starší desktop verzi systému. Během praxe bylo mým hlavním úkolem mimo jiné připravit uživatelská data k importu tak, aby při aktualizaci systému zákazníkovi zůstalo stejné nastavení, jako měl v dosavadní verzi systému.

## 4 Řešení zadaných úkolů

K řešení zadaných úkolů jsem si vybrala skriptovací jazyk Python [1] a prostředí Jupyter Notebook [2], což je webová verze pythonovské konzole. Jupyter poskytuje možnost postupného ladění a vizualizace dat. Python nám umožňuje pracovat s velkými datovými sadami, vizualizovat data pomocí různých modulů, podporuje mnoho modulů pro strojové učení a poskytuje spoustu dalších výhod při analýze dat. Většina dat ve firmě Atlas consulting se ukládá jako JSON [3], což je odlehčený a na jazyce zcela nezávislý formát pro výměnu dat. JSON je jednoduše čitelný i zapisovatelný člověkem a snadno analyzovatelný i generovatelný strojem.

Pro téměř všechny skripty jsem využila modulu *Pandas*, který mimo jiné umí přeformátovat data z JSON objektu do přehledných tabulek. Umožňuje pracovat s daty v ucelené struktuře podobné relačním databázím. Díky mnoha zabudovaným funkcím umí rychle zpracovat velké množství dat, vyhodnocovat je a čistit, čímž šetří mnoho času. Dalším modulem, bez kterého se analytik pracující v Pythonu neobejde, je *NumPy*. Ten definuje typ pro n-rozměrné pole (nejčastěji čísel) a API pro práci s takovým polem. Téměř všechny knihovny, kde se objevují větší matice či tabulky (jako například *Pandas*), jsou buď postavené na *NumPy*, nebo jej podporují.

Samozřejmostí při zpracovávání dat z JSON objektů je také stejnojmenný modul, který poskytuje funkce pro konverzi dat mezi JSON objekty a pythonovským dictionary (v češtině tzv. slovník). Jelikož data se ukládají na interní server pod URL adresu své využití ve skriptech našel také modul *urllib*. Při analýze dat hraje důležitou roli datum a čas záznamů, pro jejich zpracování využíváme modul *time* a *datetime*.

### 4.1 IQLink-reporty

Aplikace IQLink [4] obecně spojuje více systémů a informace o jejich využití prezentuje konzistentně v jednom snadno použitelném rozhraní. Ve společnosti Atlas consulting sbírá informace o využití systému CODEXIS a prezentuje je ve formě JSON objektů. Data se ukládají v časové linii. Ačkoliv součástí URL adresy, pod kterou jsou uložena, je kromě data i čas, lze je zobrazit pouze po dnech. Tím pádem z adresy

`http://.../rest/events/getEventsJSON/2018-09-03T00:00:00/2018-09-04T00:00:00`

načteme data ze dne 3. 9. 2018 i ze dne 4. 9. 2018 a ne pouze data za 24 hodin od 3. 9. 00:00:00 do 4. 9. 00:00:00. Stejná data získáme i tehdy, pokud by nás napadlo zadat například interval jedné hodiny během půlnoci:

`http://.../rest/events/getEventsJSON/2018-09-03T23:30:00/2018-09-04T00:30:00`

Množství nasbíraných dat se pohybuje v řádech deseti až statisíců záznamů denně. Každý jednotlivý záznam je přehledně rozčleněn a obsahuje obvykle okolo deseti řádků. Takovou spoustu dat není na běžném počítači schopný najednou načíst žádný prohlížeč.

#### 4.1.1 Reporty o využití systému

Nejdříve bylo nutné u každého skriptu pochopit, co vlastně zpracovává. K žádnému nebyla vytvořena žádná dokumentace ani stručný popis o tom, co dělá nebo co konkrétně je výstupem. Vzhledem k tomu, že jsem se s jazykem Groovy [5] setkala poprvé, šlo mi nejdřív pochopení kódu pomalu, ale čím více jsem jich prošla, tím to bylo jednodušší. Nakonec se mi podařilo funkčnost všech skriptů rozluštit a vytvořit je nově v interaktivním Jupyteru.

Jednotlivé skripty jsou volány pomocí utility *curl*. Součástí příkazu je kromě skriptu URL adresa, kde jsou uložena data ve formě JSON objektů. Ta však není úplná a při samotném otevření prohlížeč žádná data nevrátí. První je tedy nutné rozparsovat příkaz a vyčíst z něj, pro které období uživatel požaduje report. Data se poté vloží do reálné adresy a po jejím zavolání již získáme data za dané období. Poté je nutné data očistit a zachovat pouze reporty o činnostech, které nás zajímají, jinak budou data zabírat zbytečně velké množství paměti, zpracování dat bude pomalé a neefektivní. Poté teprve může začít zpracovávání dat a počítání statistik. Každý výstup musí být ve správném formátu, ať už se má zobrazit uživateli přímo nebo má být uložen do textového souboru či CSV souboru.

Zde je příklad zpracování dat jednoho z nejčastěji používaných skriptů:

---

```
for index, row in df.iterrows():
    if row['__type'] == "cz.atlascon.iqlink.AuthLogin":
        loginCount += 1
        uniqueUserLogin.append(row['userId'])
        userSim.append(row['userId'])
        userSim = list(set(userSim))

        if len(userSim) > loginMax:
            loginMax = len(userSim)

    if row['__type'] == "cz.atlascon.iqlink.AuthLogout":
        if row['userId'] in userSim:
            userSim.remove(row['userId'])

    if row['__type'] == "cz.atlascon.iqlink.AuthError":
        loginRefused += 1

uniqueUserLogin = list(set(uniqueUserLogin))
uniqueUserLogin = str(len(uniqueUserLogin))
```

```

d = day.strftime("%Y%m%d")

loginCount = str(loginCount)
loginMax = str(loginMax)
loginRefused = str(loginRefused)

rslt += (d + "," + loginCount + "," + uniqueUserLogin + ",")
rslt += (loginMax + "," + loginRefused + "\n")

```

---

Výpis 1: Ukázka části skriptu pro IQLink-reports

Pomocí výpisu 1 lze zjistit, kolik bylo v daný den přihlášení, kolik jednotlivých uživatelů se přihlásilo, kolik nejvíce uživatelů bylo v jednu chvíli přihlášeno a kolika uživatelům se přihlásit nepodařilo. Tato část kódu je součástí cyklu `for`, který prochází požadované dny, a vypočítává jednotlivé hodnoty z již očištěného pole `dat`. Příklad výstupu `dat` je zachycen na obrázku 1.

```

20180801,110,85,29,0
20180802,83,60,23,0
20180803,52,40,20,1
20180804,0,0,0,0
20180805,0,0,0,0
20180806,88,61,23,0
20180807,87,71,27,1

```

Obrázek 1: Výstup reportu

Nad tímto úkolem jsem strávila zhruba 10 pracovních dní. Byl to pro mne vhodný start ve firmě. Pomohl mi seznámit se s vyvíjeným softwarem a interními daty.



#### 4.1.2 Reporty o uživatelích

Mým druhým úkolem bylo vytvoření reportů o činnosti jednotlivých uživatelů. To znamenalo roztrždit data z jednoho JSON objektu obsahujícího data o činnosti všech uživatelů do několika menších JSON objektů, zvláště pro každého uživatele. Aplikace IQLink sbírá informace z IS CODEXIS od roku 2014, ale teprve od konce srpna roku 2016 ukládá také uživatelské ID vztahující se k jednotlivým událostem.

Samotné vytvoření skriptu nebylo nijak zvláště složité. Časově náročné bylo zpracování tak velkého množství dat. Kvůli omezené kapacitě paměti nebylo možné načíst všechna data, roztrždit je podle uživatelů, seřadit podle období a vygenerovat příslušný JSON objekt pro jednotlivé uživatele. Vzhledem k množství a velikosti dat lze udržovat v paměti pouze data za jeden den. Jedině tak zachováme efektivitu zpracování.

Nejprve je nutné si vytvořit seznam dní (dat) od 25. 8. 2016 do předešlého dne - záznamy ze dne spouštění skriptu ještě nejsou kompletní. O to se postará funkce *date\_range* modulu *Pandas*, jejímiž vstupními parametry je datum začátku a datum konce požadovaného intervalu. Poté cyklem *for* načítáme data z jednotlivých dní do paměti a zpracováváme do jednotné podoby. Jelikož každý systém, který chce zůstat aktuální, se neustále vyvíjí, tak jednotlivé bloky dat neobsahovaly všechny atributy jako nyní. Tudíž bylo potřeba nejprve zajistit sjednocení bloků obsahujících informace o stejných činnostech uživatelů IS CODEXIS. Během konverze z JSON objektu do datového frameworku knihovny *Pandas* se mění některé datové typy. Například z celých čísel se stala čísla reálná či *NumPy.Int*. Přetypování nám komplikuje fakt, že atributy bez hodnoty se zobrazují jako *NumPy.NaN*, jež nelze přetypovat například na obyčejný *integer*. Zpracování každé hodnoty je potřeba obalit výjimkou. Binární hodnota se ve stávajících objektech ukládá jako 1 nebo 0, což *Pandas* vidí jako číslo. To způsobilo, že například atribut *isFavorite* nabýval hodnot 1.0 nebo 0.0. Pro všechny podobné případy byly číselné hodnoty nahrazeny booleovskými hodnotami *True* a *False*. Komplikace se objevily také u textových hodnot. V některých textových řetězcích se objevovaly speciální znaky, jež Python nedokázal přečíst.

Jakmile byla data připravena a uložena do pythonovského slovníku pod klíčem představujícím uživatelské ID, mohla začít konverze na JSON objekt a ukládání do souborů. Ukázku kódu ukládání dat do souborů lze vidět ve výpisu 2. Pro uživatele, kteří již mají vytvořený soubor, se daný soubor otevře nejprve v režimu *rb+*. Tento režim umožňuje čtení a zápis v binárním formátu. Pomocí funkce *SEEK* nalezneme poslední znak v souboru - v našem případě znak *]*, jež uzavírá uložený JSON objekt a pomocí funkce *TRUNCATE* jej odstraníme. Následně soubor otevřeme v režimu *a* neboli *append* a zapíšeme záznamy z daného dne. Pro uživatele, kteří dosud nemají založený soubor, se vytvoří nový soubor pomocí módu *w* neboli *write*. Pokud bychom otevřeli již vytvořený soubor v režimu *w*, přepsali bychom veškerý obsah.

Nakonec se podařilo skript odladit, všechna data za necelé 3 roky zpracovat a vytvořit reporty o využívání systému jednotlivými uživateli. Vytváření skriptu trvalo necelé 3 dny, zpracování dat a vytváření reportů pak další skoro 2 týdny.

---

```
data = json.dumps(record, ensure_ascii=False, indent=4)
if row['userId'] not in usersId:
    usersId.append(user)
    with open(
        '/tmp/export/Reports_per_user/' + user + '.json',
        mode = 'w',
        encoding="utf-8"
    ) as report:
        report.write('[' + data + ']')
else:
    with open(
        '/tmp/export/Reports_per_user/' + user + '.json',
        mode = 'rb+'
    ) as report:
        report.seek(-1, os.SEEK_END)
        report.truncate()
    with open(
        '/tmp/export/Reports_per_user/' + user + '.json',
        mode = 'a',
        encoding="utf-8"
    ) as report:
        report.write(',' + data + '']')

```

---

Výpis 2: Ukázka zápisu uživatelských dat do souborů

## 4.2 Zpracování uživatelských profilů z online verze IS

Informační systém CODEXIS je vyvíjen ve dvou základních verzích - desktop a online. Obě verze čeká v brzké době rozsáhlá aktualizace. Mým úkolem bylo nachystat uživatelská data k importu tak, aby uživatel po aktualizaci zakoupeného softwaru mohl pokračovat tam, kde skončil a neztratil při tom veškeré své nastavení.

Jako první by měla nastat aktualizace online verze IS. Veškerá data využívaná v této verzi jsou uložena v databázi MySQL [6]. Když jsem na tomto úkolu začala pracovat, bylo mi pouze řečeno, abych připravila data. Zjednodušeně řečeno vzít všechna data o uživateli z databáze a do souboru uložit vytvořené JSON objekty, které se poté naimportují do nové verze systému.

Pro připojení k databázi z Jupyteru jsem využila knihovnu *PyMySQL* a pro zobrazení dotazů funkci `read_sql_query` z knihovny *Pandas*. Po prvotní analýze tabulek v databázi jsem dospěla k závěru, že nejefektivnější bude načítat data po jednotlivých tabulkách a ukládat je do pythonského slovníku pod jednotlivé uživatele, kde klíč ve slovníku byl zároveň uživatelským ID v databázi. Pochopitelně bylo vhodné začít tabulkou `user_list`, která obsahuje všechny aktivní uživatele, jejich údaje a základní nastavení. Tato tabulka obsahuje 32 atributů, například jestli v případě změny u sledovaných dokumentů, chce být uživatel informován e-mailem. Data nebylo nijak zvlášť složité z databáze uložit do slovníku, bylo pouze nutné zajistit konzistenci dat. Třeba hodnoty, které nabývají pouze dvou hodnot, se do databáze ukládají různým způsobem. Některé binárně, jiné logicky pomocí hodnot `True/False` a jiné jako celé číslo 0 nebo 1, které si *Pandas* opět přetypoval na číslo reálné a zobrazoval je jako 0.0 či 1.0. Binární data se v tabulkách zobrazovala jako `b"\x01"` pro `True` a `b"\x00"` pro `False`. Po dohodě s kolegy jsme se rozhodli, že všechny tyto hodnoty změníme na jednoznačné `True/False`.

Poté mohlo začít navazování dalších tabulek obsahujících uživatelské ID: poslední otevřené dokumenty, oblíbené a sledované dokumenty, oblíbené a sledované normy, historie vyhledávání atp. To už šlo docela snadno, až na jednu nemilou skutečnost, která se objevuje u většiny rozsáhlých databází: jakmile uživatel neobnoví licenci, jeho profil je vymazán z tabulky `user_list`, avšak jeho uložené nastavení, historie procházených dokumentů apod. v ostatních tabulkách databáze zůstává. Tato skutečnost byla vyřešena jedinou jednoduchou podmínkou a export dat do souborů v podobě JSON objektů byl hotov.

Po třech měsících, během kterých jsem pracovala na jiných úkolech (mj. i na přípravě dat z desktop verze IS), přišel požadavek od kolegů z backendu na zpracování stejných dat do jimi připravených schémat. Tím se moje předchozí práce stala mírně zbytečnou, alespoň jsem však nyní měla připravenou většinu dat a už stačilo jen některá z nich naformátovat do příslušných schémat.

Připravena byla 4 schémata a všechna měla základní strukturu stejnou:

```
{"schemaName": název schéma,  
  "event": schéma eventu}
```

Jednotlivá schémata jsou vlastně kombinací několika menších schémat a předdefinovaných atributů. Pro představu schéma pro otevřené a oblíbené dokumenty vypadá následovně:

```
record cz.atlascon.app.history.DocEvent {
    cz.atlascon.core.DateTime eventTime;
    cz.atlascon.core.DocId docId;
    cz.atlascon.app.history.DocEventType eventType;
}
```

Jednotlivé části schématu:

- `cz.atlascon.core.DateTime eventTime` se skládá z atributů typu *integer*: `year`, `month`, `day`, `hour`, `minute`, `sec`
- `cz.atlascon.core.DocId docId` se skládá z atributu `docId` typu *integer* a atributu `dz` typu *enum*, který musí nabývat jedné z těchto hodnot: CR, LT, EU, ES, JD, VS, SK
- `cz.atlascon.app.history.DocEventType eventType` je atribut typu *enum* a nabývá hodnot OPEN, FAVORITE, UNFAVORITE

Výstupem je JSON objekt, jehož ukázkou lze vidět na obrázku 2.



```
1  {
2    "schemaName": "cz.atlascon.app.history.DocEvent",
3    "event": {
4      "eventTime": {
5        "year": 2019,
6        "month": 4,
7        "day": 3,
8        "hour": 10,
9        "minute": 54,
10       "sec": 14
11      },
12      "docId": {
13        "docId": 5008,
14        "dz": "CR"
15      },
16      "eventType": "FAVORITE"
17    }
18  }
```

Obrázek 2: Blok dat připravený k importu

Nyní máme připravené bloky dat k importu do testovacího prostředí nové verze softwaru. K importu jsem využila funkci `post` z knihovny *Request*. Po odladění pár drobností jsou uživatelská data z online verze zcela připravena k importu do nové verze softwaru. Celkový čas strávený nad tímto úkolem bych odhadla na 5-10 pracovních dní.

### 4.3 Zpracování uživatelských profilů z desktop verze IS

Stejný výsledek, kterého jsme dosáhli u dat z online verze, je požadován i pro starší verzi systému CODEXIS. Data z něj se ukládají do databáze Firebird [7]. Některá data jsou uložena hromadně v binární podobě. Jednodušší proto je využít možnosti zálohování profilů ze softwaru. Záloha se provede tak, že je vygenerován UPB soubor, což je obdoba CSV souboru a lze jej také otevřít ve většině tabulkových editorů. Pomocí něj musíme daný soubor uložit jako CSV soubor, aby byl pro Python čitelný. Po nahrání do Jupyteru si data zobrazíme do tabulky pomocí modulu *Pandas*. Srovnání zobrazení dat v databázi Firebird pomocí příkazového řádku, ve vygenerovaném souboru pomocí tabulkového editoru a pomocí Pythonu v Jupyteru je možné vidět na obrázcích 3, 4 a 5. Na všech obrázcích je zobrazen stejný záznam o sledovaném souboru.

```
Příkazový řádek - py
>>> cur.execute('select * from usr_history where recid = 1205763')
<fdb.fbcore.Cursor object at 0x0263CD70>
>>> cur.fetchall()
[]
>>> cur.execute('select * from usr_docs where recid = 1205763')
<fdb.fbcore.Cursor object at 0x0263CD70>
>>> cur.fetchall()
[(1205763, -1200040, -2, 85665, None, None, b'\xa1N\x01\x00\x00\x00\x00\x0eD\xe5@\x02\x00\x00\x00PD\xc7\x00\x00\x00199/2017 Sb. Z\u0xe1kon, kter\u0xfadm se m\u0xecn\u0xed z\u0xe1kon \xe8. 361/2000 Sb., o provozu na pozemn\u0xedch komunikac\u0xedch a o zm\u0xecn\u0xe1ch n\u0xeackter\u0xfdch z\u0xe1kon\u0xf9 (z\u0xe1kon o silni\u0xe8n\u0xedm provozu), ve zn\u0xecn\u0xed pozd\u0xecj\u0xa9\u0xedch p\u0xf8edpis\u0xf9, z\u0xe1kon \xe8. 247/2000\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00', None, None)]
```

Obrázek 3: Firebird

20	tracked;1205763;1200040;85665;PD;43559;MTk5LzlwMTcgU2luIjFrha29uLCBrdGVy/W0gc2Ugbexu7Sb64
21	favourite;1205759;1200040;5462;PD;"361/2000 Sb. Zákon o provozu na pozemních komunikacích a o zr
22	favourite;1205761;1200040;109717;PD;"241/1939 Sb. Nařízení o jízdě motorovými vozidly"
23	favourite;1205773;1200040;146238;JD;"Usnesení - Nedovolená výroba a držení omamných a psychotr
24	highlighter;1205757;1200040;13986;AgAAADeZOTg2OzEuMS4yMDE5OzgyMzlwOzgyNDk3OzM5MDszO
25	highlighter;1205778;1200040;3242141;AgAAADMynNDIxNDE7OzE0MDY7MTQxNjs4Ozg7MDsxOzsKMzlw
<div>⏪ ⏩ 🔍 LESNICAKOVA 📄</div>	
Připraven	

Obrázek 4: Soubor UPB se zálohou

0	recid	user_id	doc_id	dat_zdroj	date	file_name	section
0	tracked	1205763	1200040	85665	PD 2019-04-04	199/2017 Sb. Zákon, kterým se mění zákon č. 36...	

Obrázek 5: Záznam v Jupyteru

Data po načtení do Jupyteru musíme nejprve dekodovat pomocí znakové sady Windows-1250 a delší znakové řetězce také pomocí kódování Base64. Speciální dekodování bylo třeba provést i pro atributy data a času. V této databázi se nevyužívá klasický UNIX timestamp [8], který

zobrazuje počet vteřin od 1. ledna 1970. Původní vývojáři systému CODEXIS si vytvořili vlastní timestamp, jež udává počet dní od 30. prosince 1988. Má-li toto číslo desetinnou část, je v něm uloženo nejen datum, ale i čas. Desetinná část reprezentuje čas v daném dni - například 0,5 je poledne a 0,25 je 6 hodin ráno.

V současné době je vytvořena funkce pro konverzi dat pro sledované, oblíbené a poslední otevírané dokumenty. Nyní pracuji na funkci, která zpracuje data sledovaných norem. Jelikož možnost sledování norem byla do systému CODEXIS implementována teprve před pár lety, není prozatím součástí zálohy uživatelského nastavení. Na doplnění této funkce a na konečném zautomatizování přípravy dat k importu do nové verze nyní pracuji spolu s týmem vývojářů, který se stará o desktop verzi systému CODEXIS.

## 5 Závěr

### 5.1 Teoretické a praktické znalosti získané během studia uplatněné v průběhu odborné praxe

V průběhu odborné praxe jsem využila celou řadu znalostí a dovedností nabytých v rámci studia. Především jsem během studia získala obecný přehled z oblasti informačních technologií. Zejména mi se mi hodila znalost jazyka Java, který je hlavním programovacím jazykem ve vývojovém týmu firmy Atlas consulting. Také jsem při svém samostudiu využila základní znalosti jazyka Python, jež je vhodný pro zpracování a analýzu dat.

Nejvíce jsem však využila znalosti z oblasti databázových systémů a dotazovacího jazyka SQL. Další cennou dovedností byla schopnost samostudia a rychlého zorientování se v aktuálně řešeném problému, kterou jsem se naučila během studia. Práce ve firmě mi přinesla řadu nových dovedností a zkušeností. Za nejcennější považuji získání zkušeností s prací v programátorském týmu.

### 5.2 Znalosti či dovednosti scházející v průběhu odborné praxe

Jelikož jsem nastupovala na praxi jakožto student teprve začínajícího třetího ročníku bakalářského studia bez předchozí praxe, chyběla mi celá řada znalostí a dovedností. Zejména jsem postrádala praktickou zkušenost s prací na reálných projektech, kterou bohužel v rámci psaní semestrálních projektů získat asi nelze. Další novou zkušeností byla práce v týmu. K tomu mi scházela zejména hlubší znalost práce s verzovacími systémy jako je Git [9].

Náš studijní obor se zaměřuje především na výuku programování, tudíž jsem postrádala také znalosti o zpracování a analýze dat. Neznala jsem ani analytické využití skriptovacího jazyka Python a prostředí Jupyter Notebook. Zcela nová pro mě byla zkušenost s databází Firebird. Tyto nedostatky jsem vyřešila načerpáním chybějících znalostí z literatury, manuálů a za pomoci kolegů.

### 5.3 Dosažené výsledky v průběhu odborné praxe a její celkové zhodnocení

Z mého pohledu byla odborná praxe pro můj budoucí profesní život velice přínosná. Práce na zadaných úkolech mě nutila se průběžně učit řadu nových zajímavých věcí. Z odborných dovedností, získaných během praxe, bych zdůraznila především získání zkušeností s vývojem a údržbou reálných systémů. Dále práci s webovou aplikací Jupyter Notebook. Zároveň bych ale chtěla zmínit i neméně důležitou oblast a to, že jsem měla možnost poznat, jak funguje společnost zabývající se vývojem rozsáhlého informačního systému a jak vypadá práce na skutečném softwarovém projektu.

Doufám, že i pro firmu Atlas consulting je moje práce přínosná a vytvořené skripty naleznou uplatnění při nasazování nové verze systému CODEXIS.

## Literatura

- [1] *The Python Standard Library* [online]. 2019 [cit. 2019-4-16]  
Dostupné z: <<https://docs.python.org/3/library/index.html>>
- [2] *Jupyter Documentation* [online]. 2019 [cit. 2019-4-16]  
Dostupné z: <<https://jupyter.org/documentation>>
- [3] *Introducing JSON* [online]. [cit. 2019-4-16]  
Dostupné z: <<https://www.json.org/>>
- [4] *IQLink* [online]. [cit. 2019-4-16]  
Dostupné z: <<http://www.iqlink.com.au/>>
- [5] *Groovy Language Documentation* [online]. [cit. 2019-4-16]  
Dostupné z: <<http://docs.groovy-lang.org/docs/latest/html/documentation/>>
- [6] *MySQL Documentation* [online]. 2019 [cit. 2019-4-16]  
Dostupné z: <<https://dev.mysql.com/doc/>>
- [7] *The Firebird FAQ* [online]. 2019 [cit. 2019-4-16]  
Dostupné z: <<http://www.firebirdfaq.org/cat3/>>
- [8] *UNIX Timestamp* [online]. 2019 [cit. 2019-4-16]  
Dostupné z: <<https://www.epochconverter.com/>>
- [9] *GitLab Documentation* [online]. [cit. 2019-4-20]  
Dostupné z: <<https://docs.gitlab.com/ee/README.html>>